Fine-Tuning BERT with Multi-Task Learning, Gradient Surgery, and Masked Language Modeling for Downstream NLP Tasks

Stanford CS224N Default Project
Mentor: Davey Huang

Gabriela Aranguiz-Dias
School of Humanities and Sciences
Stanford University
gadias@stanford.edu

Janelle Cheung
Department of Computer Science
Stanford University
janellecheung@stanford.edu

Abstract

Fine-tuned transformer-based language models are incredibly powerful tools that have revolutionized natural language processing, but this fine-tuning is not always generalizable across different tasks. Multi-task learning allows us to potentially create models that can perform well on several different tasks with the same data, transferring the knowledge learned from one task to another. Our goal in this project was to leverage the output embeddings of the Bidirectional Encoder Representations Transformers (minBERT) model, implement multitask fine-tuning and key extensions, and investigate how multitask learning and these extensions perform well on the following three downstream tasks: sentiment analysis, paraphrase detection, and semantic textual analysis. The main questions we attempt to answer with our project are as follows: (1) How does fine-tuning our implementation of BERT with distinct modifications differ from using exclusively multi-task fine-tuning as an extension to BERT? (2) Do these methods lead to better or worse performance on the selected tasks? (3) How can this inform best practices regarding multitask learning for language models moving forward? Therefore, as extensions, we built in the following to our multitask fine-tuning implementation, and compared all results against multitask fine-tuning and single-task fine-tuning as baselines: gradient surgery (Yu et al. (2020a)), Masked Language Modeling (MLM) (Liu et al. (2019)), and Cosine-Similarity Fine-Tuning (Reimers and Gurevych (2019)). We found that multitask fine-tuning with gradient surgery yielded the best performance, followed by regular multitask fine-tuning. We believe that this is because gradient surgery alleviates the negative impact of conflicting gradients between tasks, leading to more stable and effective learning across all tasks. More research should be done to further optimize gradient surgery techniques and investigate other strategies for addressing task interference, with the goal of improving multitask learning performance and better leveraging shared knowledge across tasks.

1 Introduction

The field of natural language processing (NLP) has undergone a significant transformation with the development of powerful pre-trained language models, such as BERT. These models have demonstrated remarkable success in a wide range of NLP tasks. However, a notable drawback of these models is their limited adaptability to multiple tasks, as being fine-tuned for a single, specific task, typically restricts their generalizability across diverse NLP challenges. To overcome this limitation, multi-task learning has emerged as a promising approach, as it allows models to be trained on multiple tasks simultaneously, facilitating knowledge transfer between tasks and yielding improved efficiency, generalization, and overall performance.

In this study, we delve into the performance of a Bidirectional Encoder Representations from Transformers (minBERT) model on three downstream NLP tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. We implement multitask fine-tuning and other extensions to the minBERT model to assess the influence of various techniques on its performance across the selected tasks. In this study, we set out to investigate several research goals. First, we aimed to examine how distinct modifications to our minBERT implementation, when fine-tuned using multi-task learning, contrast with employing only multi-task fine-tuning as an extension to BERT. Second, we sought to assess the impact of these modifications on the performance of sentiment analysis, paraphrase detection, and semantic textual similarity tasks. Lastly, our high-level objective was to understand how the outcomes of this study can contribute to the establishment of best practices for multi-task learning in language models.

To address these questions, we extend our multi-task fine-tuning implementation by integrating gradient surgery (Yu et al. (2020a)), additional Masked Language Modeling (MLM) pretraining Liu et al. (2019) on an additional dataset, and Cosine-Similarity Fine-Tuning techniques, and benchmark the results against multi-task and single-task fine-tuning baselines. Our experiments employ the Stanford Sentiment Treebank dataset Socher et al. (2013) for sentiment analysis, the Quora dataset for paraphrase detection, and the SemEval STS Benchmark dataset Agirre et al. (2013) for semantic textual similarity. We appraise our model's performance using test set accuracies for paraphrase detection and sentiment analysis tasks, and Pearson correlation score for semantic textual similarity.

Our findings reveal that multitask fine-tuning with gradient surgery provides the best performance, addressing our research goals by illustrating the efficacy of various modifications when applied to our minBERT implementation in a multi-task learning setting. These results demonstrate the differences in performance when comparing our distinct modifications with solely employing multi-task fine-tuning as an extension to BERT. Furthermore, the outcomes provide valuable insights into the effectiveness of these techniques in enhancing the performance of language models across sentiment analysis, paraphrase detection, and semantic textual similarity tasks. Future research endeavors should concentrate on preventing overfitting and honing on on generalizability, building upon the insights gained from this study to further improve and refine multi-task learning approaches.

2 Related Work

In our research, we build upon two influential papers, "Gradient Surgery for Multi-Task Learning" (Yu et al. (2020a)) and "RoBERTa: A Robustly Optimized BERT Pretraining Approach" (Liu et al. (2019)). The "Gradient Surgery for Multi-Task Learning" paper (Yu et al. (2020a)) primarily focuses on addressing the issue of gradient interference in multi-task learning. Gradient interference occurs when gradients from different tasks conflict with each other, causing the learning process to be less effective than training the tasks independently. The authors identify a "tragic triad" of conditions that lead to this issue: conflicting gradients, high positive curvature, and a large difference in gradient magnitudes. To mitigate the effects of gradient interference, the paper proposes to project conflicting gradients onto the normal plane of the other, which prevents the interfering components from being applied to the network. This specific method is referred to as "projecting conflicting gradients" (PCGrad). The paper demonstrates that applying PCGrad to various multi-task learning scenarios leads to improved performance compared to standard multi-task learning approaches.

The "RoBERTa: A Robustly Optimized BERT Pretraining Approach" paper (Liu et al. (2019)) presents an improved version of the BERT model called RoBERTa, which places a stronger emphasis on masked language modeling (MLM) as the primary pre-training task. The authors of the paper recognize that the MLM objective is crucial for learning meaningful language representations. They make several key adjustments to the original BERT's pretraining process to enhance the model's performance on the MLM task. Among these modifications are increasing the batch size for training, using longer training sequences, and dynamically changing the masking pattern during training. Additionally, they remove the next sentence prediction (NSP) task, as it was found to have a limited impact on downstream performance. By concentrating on optimizing the MLM task, RoBERTa achieved state-of-the-art performance on a range of NLP benchmarks, outperforming the original BERT model and other transformer-based models at the time of publication.

Our project integrates the ideas from both the "Gradient Surgery for Multi-Task Learning" (Yu et al. (2020a)) and the "RoBERTa: A Robustly Optimized BERT Pretraining Approach" (Liu et al. (2019))

papers within a multi-task learning framework. We leverage the ideas described in RoBERTa's enhanced masked language modeling capabilities to provide a strong pretraining foundation for our model. We then apply the gradient surgery technique from Yu et al. (2020a) to mitigate gradient interference issues that may arise when training on multiple NLP tasks simultaneously. This combination allows us to investigate the performance of our model on various NLP tasks, drawing upon the strengths of both approaches to improve the efficiency and effectiveness of multi-task learning in deep learning models.

3 Approach

In this section, we detail our implementations of the minBERT model, the baseline models, and of the various paths taken to improve multitask learning performance.

3.1 Implementation of minBERT

We began by implementing the minBERT model, which involved implementing the multi-head attention layer of the BERT transformer and realizing the entire BERT transformer layer and implementing the `step()` function of the AdamW optimizer.

3.2 Baseline Model

Our baseline model was implemented as a multi-task predictor for sentiment analysis, paraphrase detection, and semantic textual similarity analysis via a round-robin method in which we do a step for each batch of each task in turn, and update the model. The model consists of minBERT layers as well as task-specific prediction heads. We append these prediction heads as final linear layers to minBERT. For the STS task, we set the output dimensions of these layers equal to the number of classes (n = 5, for each classifying sentiment). We then take the predictions to be the index of the largest outputted logit. For the paraphrase detection task, we output a single logit corresponding to how similar the inputs are. For the regression task (based on the SemEval dataset), we simply take the value of the final logit to be the prediction. All prediction heads are trained with dropout, and all hyperparameters used are described here. We used cross-entropy loss for the paraphrase detection and sentiment analysis tasks and mean squared loss for the semantic textual evaluation task.

3.2.1 Single-task Models

To establish more context for baseline performance, we also implemented single-task fine-tuning on minBERT, for all three tasks. This allowed us to asses the task-specific benefits and drawbacks of fine-tuning minBERT across multiple datasets simultaneously. We reused our implementations of each prediction head to train on each respective task alone.

3.2.2 Gradient Surgery

We addressed the issue of conflicting gradients in multitask learning by implementing gradient surgery using PCGrad, a technique to mitigate negative transfer between tasks (Yu et al. (2020b)). This method projects the gradient of the i-th task g_i onto the normal plane of another conflicting task's gradient g_j, depicted mathematically below.

$$g_i = g_i - \frac{g_i \cdot g_j}{||g_j||^2} \cdot g_j \tag{1}$$

We used the PyTorch implementation provided by Tseng (2020). This required zipping up the dataloaders for all three tasks, calculating one loss, and taking one step per batch in order to manipulate the gradients of all three tasks at the end of each batch, instead of using round-robin multitask training. In other words, for this experiment, we were restricted in keeping the batch sizes for all three tasks the same.

3.2.3 Cosine-Embedding Loss

For the semantic textual similarity task, we experimented with using cosine-embedding loss instead of mean squared loss, following the precedent in the literature (Reimers and Gurevych (2019)). The

cosine similarity between two vectors, A and B, is calculated as:

$$\texttt{cosineSimilarity}(A, B) = \frac{A \cdot B}{|A|\,|B|} \tag{2}$$

For cosine-embedding loss, the similarity between the two sentence embeddings is compared to a target similarity value, which we derive from our ground-truth batch labels in the STS Benchmark dataset. We implemented this via a class and used the methods within it in our experiments.

3.2.4 Additional Pre-training with Masked Language Modeling

To improve performance on the downstream tasks, we conducted additional pre-training with masked language modeling (MLM) on the IMDb movie review dataset, which matched the domain of our sentiment analysis task. MLM enables the language model to learn general patterns in language and build rich representations of word relationships.

3.2.5 Addressing Overfitting

We observed overfitting in our models and adjusted some hyperparameters to improve generalizability: (1) Increased batch sizes to provide smoother gradient steps and more training examples per batch and (2)Increased dropout probabilities when training with the prediction heads to force the model to learn more robust and generalizable features.

3.2.6 Large Batch Sizes and Gradient Accumulation in Pre-training

We implemented gradient accumulation to enable larger effective batch sizes during pre-training for learning more general language patterns. This approach allowed GPU space efficiency while providing the benefits of large batch sizes during MLM pre-training.

3.2.7 Experimenting With Hyperparameters

We implemented several hyperparameter changes to mitigate overfitting to the following variants of multitask fine-tuning: basic multitask fine-tuning and multitask fine-tuning with additional-pretraining with masked language modeling.

4 Experiments

4.1 Data

To obtain a pre-trained model, we utilized the BERT-base-uncased model as our foundation. This pre-trained model employs a masked language modeling objective, resulting in a robust pre-trained model for the English language with 110 million parameters ([1]).
For our experiments, we used three primary datasets, each corresponding to a specific task, described below in full:

> **Sentiment Analysis:** We employed the Stanford Sentiment Treebank (SST) dataset for this task (Socher et al. (2013)). The dataset comprises 11,855 single sentences extracted from movie reviews. These sentences were parsed into 215,154 unique phrases and labeled by human judges as negative, somewhat negative, neutral, somewhat positive, or positive. The input for this task consists of a sentence, and the output is a sentiment label ranging from negative to positive.
>
> **Paraphrase Detection:** For paraphrase detection, we used the Quora dataset, which consists of 400,000 lines of potential question duplicate pairs([2]). The task involves determining if a pair of questions are paraphrases of one another. The input for this task is a pair of questions, and the output is a binary label indicating whether the questions are paraphrases (1) or not (0).

[1] https://huggingface.co/bert-base-uncased

[2] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Semantic Textual Analysis: We employed the SemEval STS Benchmark dataset for this task (Agirre et al. (2013)). The dataset includes 8,628 different sentence pairs with varying degrees of similarity. The task seeks to measure the degree of semantic equivalence between pairs of sentences on a scale from 0 to 5. The input for this task is a pair of sentences, and the output is a continuous similarity score ranging from 0 (completely dissimilar) to 5 (semantically equivalent).

4.2 Evaluation method

In our evaluation, we use distinct metrics tailored to each downstream task. The metrics are compared to the performance of our model on the validation and test sets when applying basic multitask fine-tuning and single-task fine-tuning, before incorporating any experimental modifications as detailed in the Approach section. For the Quora and SST datasets, which correspond to the paraphrase detection and sentiment analysis tasks, we measure the accuracy of the model on the test set. Accuracy is defined as the proportion of test set instances that are correctly classified by the model, providing a quantitative assessment of the model's performance in these tasks. For the SemEval dataset, which is used for the semantic textual analysis task, we evaluate the model's performance using the Pearson Correlation score (Agirre et al. (2013)). This metric quantifies the degree of linear relationship between the predicted and the ground truth similarity scores, offering an appropriate evaluation method for tasks that involve continuous output values. By utilizing these metrics, we aim to provide a comprehensive and task-specific assessment of our model's performance across the diverse NLP tasks under consideration.

4.3 Experimental details

For multi-task pretraining, we utilized the BERT-base-uncased pre-trained model, freezing the BERT weights and layers. The hyperparameters for multi-task pre-training included:

- Batch size: 8
- Dropout rate: 0.3
- Number of epochs: 10
- Learning rate: 1×10^{-3}

For multi-task fine-tuning, we unfroze the BERT weights and layers, updating them alongside the prediction heads for each task. The hyperparameters were identical to those used in the multi-task pre-training above, with the exception of the learning rate, which was set to 1×10^{-5}. For all experiments, including cosine-embedding loss, gradient surgery, and the first pass of MLM, we also unfroze the BERT weights and applied the same hyperparameters as in multi-task fine-tuning for the three downstream tasks. For additional pre-training with MLM, we also learned an MLM prediction head, which was disregarded during downstream fine-tuning. Our first version of our MLM implementation used the following hyperparameters:

- Pre-training batch size: 4
- Pre-training number of epochs: 3

Our second version had the same number of pre-training epochs but employed:

- Pre-training batch size: 4
- Pre-training gradient accumulation step: 32

To combat overfitting, we later experimented with adjusting the following hyperparameters for multi-task fine-tuning to these values:

- Batch size: 64
- Dropout rate: 0.5

These changes were also applied to the second version of our MLM model.

4.4 Results

Test Leaderboard Scores

Sentiment	Paraphrase	Similarity	Average
0.524	0.649	0.326	0.500

Baseline Model

Model	Sentiment	Paraphrase	Similarity	Average
Multitask-pretrained	0.302	0.377	0.225	0.301
Multitask-finetune*	0.499	0.650	0.351	0.500

* Baseline model moving forward

Single-Task Models

Model	Pretrain	Finetune
Sentiment analysis	0.596	0.743
Paraphrase detection	0.378	0.791
Semantic textual similarity analysis	0.226	0.352

Extensions on Baseline Model

Model	Sentiment	Paraphrase	Similarity	Average
Gradient surgery	0.669	0.487	0.380	0.512
Cosine-embedding loss	0.142			
Additional pre-training with MLM	0.416	0.634	0.289	0.446
Adjusted hyperparameters	0.466	0.579	0.337	0.461
Adjusted hyperparameters and additional pre-training with MLM	0.357	0.727	0.293	0.459

Gradient surgery: We observed a slight improvement in the average performance by implementing multitask fine-tuning with gradient surgery as PCGrad (Yu et al. (2020a)). The sentiment accuracy increased by 0.179, the paraphrase accuracy decreased by 0.163, and the similarity correlation increased by 0.029. The average performance increased by 0.012. This improvement was smaller than what we expected, as we were hoping for a more significant improvement.

Cosine-embedding loss: Implementing cosine-embedding loss for sentiment analysis resulted in a significant decrease in performance on the similarity task. The similarity correlation decreased by 0.209, and both the training and dev accuracies were low across epochs. The final training accuracy decreased by 0.781, and the final dev accuracy decreased by 0.336. Thus, cosine-embedding loss was detrimental to our model and semantic similarity task, which was not what we expected given the existing literature.

Additional pre-training with MLM: We observed a slight decrease in performance across all three tasks by conducting additional pre-training with MLM. The sentiment accuracy decreased by 0.083, the paraphrase accuracy decreased by 0.016, and the similarity correlation decreased by 0.062. The average performance decreased by 0.054. These results were not drastically negative, but we had originally hypothesized an improvement in performance.

Adjusting hyperparameters: Adjusting the hyperparameters (batch size and dropout) resulted in a decrease in performance across all three tasks. The sentiment accuracy decreased by 0.033, the paraphrase accuracy decreased by 0.071, and the similarity correlation decreased by 0.014. The average performance decreased by 0.039. We also observed that the training accuracy was increasing significantly while the dev accuracy remained low, indicating that we were still overfitting to the training data. These results were very unexpected given that our model was seeing about eight times as much data per batch but overfitting even more than the baseline model.

Additional pre-training with MLM and adjusted hyperparameters: Conducting additional pre-training with MLM and adjusted hyperparameters resulted in a decrease in performance across all three tasks. The sentiment accuracy decreased by 0.142, the paraphrase accuracy decreased by 0.077,

and the similarity correlation decreased by 0.058. The average performance decreased by 0.041. The final accuracies were similar to the baseline model, indicating that this experiment did not make our model more generalizable. Given that adjusting the fine-tuning hyperparameters did not improve the model, these results were not too unexpected. However, we had hoped that by pre-training with effective batch sizes of 128, a value thirty-two times greater than the first batch size, we might see some slight improvement in performance.

The following Analysis section provides an explanation for each of the experimental results obtained and outlines the reasons why we believe we achieved those results.

5 Analysis

All of our extensions were developed by building on top of the baseline model, a multi-task fine-tuning model that fine-tunes a pre-trained BERT model and three task prediction heads by training on each of the three tasks in a round-robin fashion. Among the extensions, gradient surgery with PCGrad provided the most significant improvement in average accuracy and correlation compared to the baseline. One possible explanation is that adversarial gradients between tasks negatively impacted the performance of each task. This is supported by the fact that our baseline multi-task fine-tuning model performed worse on the three tasks compared to models fine-tuned exclusively on each task. Thus, it is seemingly the case that gradient updates improve accuracy on some tasks and degrade the model's accuracy in other tasks. Though gradient surgery addresses some of these issues, we observed that the model fine-tuned on all three tasks using gradient surgery still exhibited inferior performance compared to the models fine-tuned and tested on single tasks. One potential explanation for this is that the model is inherently limited in its capacity to absorb information about each task. As far as our failure to observe the theoretical benefits of transferring knowledge between tasks, it may be the case that the three tasks at hand tested sufficiently different aspects of language understanding such that we were not able to observe this benefit.

Surprisingly, replacing mean squared loss with cosine-embedding loss for the semantic textual similarity task within the multi-task context did not yield improved results. In fact, it considerably degraded the similarity task performance. This is unexpected, as cosine-embedding loss is primarily designed for semantic textual similarity tasks, focusing on the angles between token vectors rather than their magnitude differences. However, the complex nature of deep learning models makes it challenging to pinpoint the exact reasons for this performance decline. One hypothesis for this performance decline is that the datasets' definition of semantic similarity might differ from the types of meanings encoded in the model's embeddings. This could cause the approach that used a separate linear layer to perform better than cosine-embedding loss, as it could learn a translation between the pre-trained BERT's representations and the representations expected by the dataset.

Additional pre-training with masked-language modeling (MLM) did not increase average performance, as we anticipated it would. Despite expecting better results for the paraphrase detection task due to the domain-specific pre-training data, our implementation of MLM was hampered by a small batch size during pre-training. We later learned that larger batch sizes are crucial for pre-training to achieve generalizable results and prevent overfitting. To address this issue, we introduced a second iteration of MLM pre-training that employed gradient accumulation and larger batch sizes. Nevertheless, this approach also failed to deliver improved performance. We believe that this lack of improvement may be due to the fact that the BERT-case-uncased model is already trained on an extensive amount of data using MLM, so our additional pre-training did not produce a significant change.

Upon examining our training losses, accuracies, and development set accuracies, we discovered that overfitting was a pervasive issue across our models, including the baseline multi-task fine-tuning model. To mitigate overfitting, we experimented with regularization by adjusting relevant hyperparameters, such as increasing batch size and dropout probability. However, these modifications neither improved average task performance nor resolved the overfitting problem (see Figure 1). These results suggest that our model requires more sophisticated and targeted regularization techniques.

Lastly, upon further inspection of the system's outputs, we observed certain patterns in the errors made by the model. For example, consider the examples in Table 1 from the Quora dataset for the paraphrase detection task below: In all three examples, the model seems to focus on the shared words and phrases between the sentences, such as "passport to go to Jamaica," "Bollywood movies," and

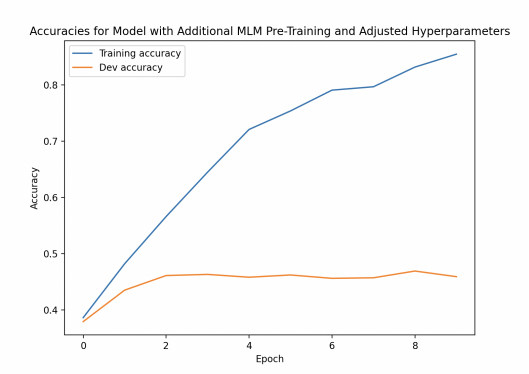

Figure 1: Accuracies for Model with Additional MLM Pre-Training and Adjusted Hyperparameters

Sentence 1	**Sentence 2**	**Prediction**
Why do I need a passport to go to Jamaica from Honduras?	Why do I need a passport to go to Jamaica from Germany?	1
Which are the worst Bollywood movies of 2016?	Which are the best Bollywood movies in 2016?	1
What is a civil war?	What is your review of Captain America: Civil War (2016 movie)?	1

"civil war." This focus on superficial similarities might have led the model to overlook the differences in meaning and context between the sentences. In tandem, the model also failed to take the words that completely change the meaning of a sentence into account, like “best” versus “worst”.

6 Conclusion

In this study, we investigated the performance of a multi-task BERT-based model on three NLP tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. We implemented several extensions, including gradient surgery, cosine embedding loss, and additional pre-training with masked language modeling (MLM) to improve the baseline model’s performance. Our main findings can be summarized as follows:

- Gradient surgery using PCGrad yielded the most significant improvement in average accuracy/correlation compared to the baseline model. This suggests that conflicting gradients between the tasks were detrimental to the individual task performance, and gradient surgery helped address this issue.
- The cosine embedding loss, although commonly used for semantic textual similarity tasks, did not improve the results for that task within the multi-task context. Instead, it dramatically decreased the performance. The reason for this unexpected outcome remains unclear, but perhaps similarity is captured differently in the minBERT embeddings versus in the STS SemEval dataset.
- Additional pre-training with MLM did not improve average performance, possibly because the pre-trained BERT model had already absorbed a considerable amount of information through MLM on a vast dataset.
- Our models, including the baseline, were likely overfitting, as evidenced by improvements in training loss and accuracy while dev accuracy remained low. Despite our efforts to implement regularization through hyperparameter tuning, the overfitting problem persisted.

Our achievements include implementing and experimenting with gradient surgery, cosine-embedding loss, and additional MLM pre-training with varying degrees of success. We also identified and attempted to combat overfitting with a variety of techniques. Although these attempts did not consistently yield the desired improvements, they provided valuable insight into challenges associated

with multi-task learning and model optimization. The primary limitations of our work, such as overfitting, the inability of our models to effectively capture subtle semantic differences, offer opportunities for further exploration and enhancement. These experiments, along with significant hyperparameter tuning, have expanded our understanding of the complexities of multi-task learning. A possible direction future work, based on our experimentation, is significantly larger datasets that could be parsed with even greater batch sizes, further increasing dropout, and exploring regularized optimization methods, such as the method proposed by Haoming Jiang and Zhao (2019), to improve downstream task performance. Another potential avenue for future research might seek to understand the sensitivity of pre-trained models to overfitting more generally; for instance, if there might be a way to finetune based on the understood semantics of a given dataset, rather than capturing its every idiosyncrasies.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

Weizhu Chen Xiaodong Liu Jianfeng Gao Haoming Jiang, Pengcheng He and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *arXiv preprint arXiv:1911.03437*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Wei-Cheng Tseng. 2020. Weichengtseng/pytorch-pcgrad.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020a. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020b. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*.